

IN THE CLAIMS

1. (Currently Amended) A method for allocating memory to a process on a computer, the method comprising:

creating a plurality of processes, each process being allocated an amount of memory, the processes including one consumer process and a donor process wherein memory allocated to the donor process is accessible to other processes~~not owned by the donor process~~; and

pooling memory of the processes together for use by the consumer process; and

hibernating the donor process in a sleep state while the allocated memory remains accessible to other processes.

2. Canceled.

3. (Previously Presented) The method of Claim 1 wherein the number of donor processes determined from the amount of allocated memory requested by the consumer process, each donor process donating allocated memory to the consumer process.

4-7. Canceled.

8. (Currently Amended) A computer program product for allocating memory to a process on a computer, the computer program product comprising a computer usable medium having computer readable code thereon, including program code which:

creates a plurality of processes, each process being allocated an amount of memory, the processes including one consumer process and a donor process, wherein memory allocated to the donor process is accessible to other processes~~not owned by the donor process~~; and

pools memory of the processes together for use by the consumer process;
and
hibernates the donor process in a sleep state while the allocated memory
remains accessible to other processes.

9. Canceled.

10. (Previously Presented) The computer progress product of Claim 8 wherein the number of donor processes is determined from the amount of allocated memory requested by the consumer process, each donor process donating allocated memory to the consumer process.

11-14. Canceled.

15. (Currently Amended) A computer system comprising:
a central processing unit;
a memory system connected to the central processing unit which:
creates a plurality of processes, each process being allocated an amount of memory, the processes including one consumer process and a donor process wherein memory allocated to the donor process is accessible to other processes~~not owned by the donor process~~; and
pools memory of the processes together for use by least one of the consumer process; and
hibernates the donor process in a sleep state while the allocated memory remains accessible to other processes.

16. Canceled.

17. (Previously Presented) The computer system of Claim 15 wherein the number of donor processes is determined from the amount of allocated memory

requested by the consumer process, each donor process donating allocated memory to the consumer process.

18-21. Canceled.

22. (Currently Amended) An apparatus for allocating memory to a process in a computer comprising:

means for creating a plurality of processes, each process being allocated an amount of memory, the processes including one consumer process and donor process; and

means for pooling memory of the processes together for use by the consumer process, the means for pooling memory further including means for donating memory, further comprising:

means for detecting a memory allocation by the donor process;

means for transferring accessibility of the allocated memory to the consumer process; and

means for deallocating the memory by sending a release request to the donor process.

23. Canceled.

24. (Currently Amended) The apparatus of Claim 22 wherein the number of donor processes is determined from the amount of allocated memory requested by the consumer process, each donor process donating allocated memory to the consumer process.

25-32. Canceled.

33. (Previously Presented) The method of Claim 1 wherein the donor process transfers ownership of allocated memory to a driver.

34. (Currently Amended) The computer program product of Claim 8 wherein the donor process transfers accessibility~~ownership~~ of allocated memory to a driver.

35. (Currently Amended) The compute system of Claim 15 wherein the donor process transfers accessibility~~ownership~~ of allocated memory to a driver.

36. (Currently Amended) The apparatus of Claim 22 wherein the donor process transfers accessibility~~ownership~~ of allocated memory to a driver.

37. (Currently Amended) The method of Claim 33 wherein the donor process hibernates in an idle state not executing program instructions~~is to caused to sleep after transferring~~ownership of allocated memory to the driver.

38. (Currently Amended) The computer progress product of Claim 34 wherein the donor process hibernates in an idle state not executing program instructions~~is to caused to sleep after transferring~~ownership of allocated memory to the driver.

39. (Currently Amended) The computer system of Claim 35 wherein the donor process hibernates in an idle state not executing program instructions~~is to caused to sleep after transferring~~ownership of allocated memory to the driver.

40. (Currently Amended) The apparatus of Claim 36 wherein the donor process hibernates in an idle state not executing program instructions~~is to caused to sleep after transferring~~ownership of allocated memory to the driver.

41. (New) A method for allocating memory to a process on a computer, the method comprising:

creating a plurality of processes, each process being allocated an amount of memory, the processes including one consumer process and a donor process wherein memory allocated to the donor process is accessible to other processes; and

pooling memory of the processes together for use by the consumer process, the number of donor processes determined from the amount of allocated memory requested by the consumer process, each donor process donating allocated memory to the consumer process, donating memory further comprising:

detecting a memory allocation by the donor process;

receiving a register command from a memory manager;

transferring ownership of the allocated memory to the memory manager in response to the register command.

42. (New) The method of Claim 37 wherein donating memory further comprises:

detecting a memory allocation by the donor process;

receiving a register command from a memory manager; and

transferring ownership of the allocated memory to the memory manager in response to the register command, the donor process responsive to the memory manager for allocating and deallocating memory accessible to a consumer process.